[0153] 2. Continuous history indicates that a continuous historical archive is automatically maintained for the Attribute.

[0154] 3. Snapshot history indicates that a snapshot historical archive is available so that a system user can periodically save a snapshot of the attribute values.

[0155] An Entity, as previously generally defined, and as noted, is a representation of a real-world object about which the system maintains information in the form of Attributes. Each Entity is identifiable by a unique ID, previously discussed as including a class ID, and is further detailed hereafter. Thus, most data can be accessed by specifying (a) the unique ID for an Entity and (b) the name of the Attribute of interest. (For some Entities, dynamic segmentation is used to store Attributes that may vary along the length of the Entity, and this specification would return a list of linear sub-sections of the Entity and values within each sub-section.)

[0156] In order to facilitate maintenance of this data, the Entities are organized into Entity classes. For example, each bridge Entity is part of the bridge Entity class. The data is organized for each Entity according to its Entity class, and applies the same maintenance practices to each Entity in the same Entity class. For example, every bridge Entity has the same collection of possible Attributes that are maintained in the same way because every bridge Entity is part of the same bridge Entity class. The Data Dictionary contains meta-data (e.g., table names, column names, whether history is maintained) about each Entity class that describe how the data associated with the Entities in that Entity class is stored and maintained.

[0157] For the exemplary embodiment, six Entity class types are implemented which define the primary characteristics of the data that can be stored in the road network system. It will be apparent to one skilled in the art that when the system and method are implemented to represent a network of sections other than for a road network, or for other applications of road networks, that different classes of Entities must be defined. A prototype table structure is defined for each Entity class type, and every Entity class with the same Entity class type uses the same table structure to contain its data. The Data Dictionary does include some Entity class parameters that will tune the prototype table structure for a particular Entity class, but the basic table structure for all Entity classes within an Entity class type is identical. It is the definition of the Entity class types and the Data Dictionary meta-data that provides flexibility and extensibility in the system by allowing development of data maintenance and presentation software that is only dependent on the six defined Entity class types, not on the potentially hundreds of specific Entity classes. The following list describes the Entity class types supported by the system of the exemplary embodiment:

[0158] **Road Network Entity Class Type.** The Road Entity Class Type refers to the objects that make up the road network. All road characteristic data is associated to the Road Entity Class Type through the Division Sections.

[0159] **Traversal Entity Class Type.** The Traversal Entity Class Type refers to linear stretches of road (e.g., streets) defined by a sequence of adjacent sub-sections of Division Sections or Road Sections. For example, interstate

highways would be implemented as a Traversal Entity Class Type. Traversals formed from Division Sections are called Anchor Traversals, and those formed from Road Sections are called Road Traversals.

[0160] Road Furniture Entity Class Type. The Road Furniture Entity Class Type refers to objects that have an associated roadway position (e.g., accidents, road improvement projects, street signs).

[0161] Spatial Object Entity Class Type. The Spatial Object Entity Class Type refers to objects that have a specific location, but whose location is not tied to the road network. In some cases a location might be tied to the road network in the real world, but the available source data provides coordinate locations for the data rather than road locations. For example, airports, counties, and schools are implemented in the Spatial Object Entity Class Type. In some cases, a collection of spatial objects might be used to populate a road characteristic value. For example, the county road characteristic value could be populated from the county spatial object county boundaries.

[0162] Non-spatial Object Entity Class Type. The Non-spatial Object Entity Class Type refers to objects that do not have any specific location associated to them. For example, bus lines could be treated in the Non-spatial Objects Entity Class Type. If an object-oriented view of this data is not desirable, the Table Entity Class Type can be used instead.

[0163] Table Entity Class Type. The Table Entity Class Type refers to tables of data for which an object-oriented view is not appropriate. For example, data